

Practical

Computational Intelligence.

Design a distributed application using the for remote computation where dient submits an integer value to the server and sonver calculates factorial and returns the result to the dient program.

The goal is to demonstrate

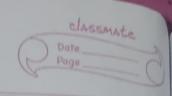
Remote Proddure (all (RPC)

in python using XML-RPC

where:

- · The client sends an integer to the server.
- · The server computes the
- . The server returns the result back to the elsent.

· Distributed computing refers to a model when different remponent of a software system are spread



across multiple machines, which work together to acheive a common god.

· Each machine in the system can act as a dient,

server or both.

- Key benifit: Task can be spuit among different systems for calability, performance and resource sharing.

-> What is RPC (kemote Procedure (all?) eff is a protocol that allows a program to execute a function/
method on another computer (server)
just like calling a local function.
The function runs on the
remote machine and the
results are sent back to the
caller.

PPC abstracts away the underlying
network communication details,
giving a feel of calling a
local function.

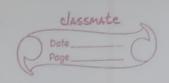
PPC supports both synchronous
and asynchronous modes.

Mhut is XMI - RPC?

· XMI-PPC is a lightweight RPC

protocol that uses:

- XMI for encuding messages



(function calls and responses).

- HTTP as the transport protocol.

. KML- RPC is simple and platform—
independent.

. Python has built-in support via:

- xmlrpc. server.

- xmlrpc. dient. -> client - Server Architecture: · Sorver: - Puns an XIYL-RPC Server on localhost: 8000. - Registers a factorial (n) function. - Listens for requests and rutarys results to dient. · Client:
- Connect to the server using Sonor Prosey. - Sends a number. - Receives and displays the factorial. -) How RPC communication Happens.

1. Client calls a function via a

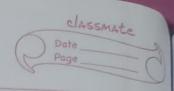
prosay object.

2. The function call is converted to XML and sent over HTTP
POST.

3. server receives it, decodes the

rigistered function.

XAM XML maps it to the



is encoded in XMI and seturned over HTTP.

s. Client decodes the XMI and and gets the recust.

This is all abstracted under the hood-you write; t likes a normal function call.

The sorver theory:

The sorver theory of input

is a non-negative

integer. If not, it raises

a value From. Errors are logged

in spe-Server. log. The sient

catches exceptions if the a server

is unreachable or input is

invalid.

Logging on the server:
· Server uses the Logging module

to write information to rpc sorverleg

· Each factorial calculation is

logged with:

- Input value.

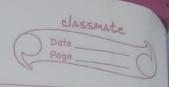
- Output result.

- Errors.

4

-> Server Threading. The server is started in · This lets it run in the rhe with while True: pars
keeps the main mogram -> Security Note (Real- World Scenario)
. In real deployments: - Use authentication and encryption (HTTPs). - Avoid escosing the server to the internet without · XML- RPC ic no not everypted by default use http://
and authentication headers
if needed. - Simplify semole interaction (like calling local function)

. Abstracts network code from developer · Language independent (dient in one · Good for modular and servicebused appl applications.



Disadvantages of RPC:
· Higher Latency than local calls.

· Hard to debug (due to retwork
fuiluers or serialization smort)

· Tight

A Code spc-server.py

1) Imorts

basic XML- RPC Server to handle dient calls.

· threading: - Used to run the server in a sep seperate thread (50 if dosent block the main

· logging: - Used for maintaining logg of all server operations

2) logging configuration:-

logging basics...

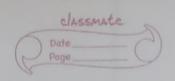
- Sets up log Rile spc-server.log.

- Logs leach message with

timestamps and message.

- Logs only INFO lever and

above (include errors too).



3) Function to compute factorial:

- · First checks if n is an integer or integer or non-negative number.
 · If not it raises error.
 · Else it computy factorial (from ton).
- · Story in recutt.
- · Logs the computation to the log file.
 · Returns the computed factorial to the elient.

exception Esception as e:

· Logs an error that occurs

during computation and returns

the error string to the

dient for debugging. start stop

computer! I python runce eschludy
is not stop value... ntl.

4) Function to start the XML-RPC Server: det start-somer(1:

· starts a somer on local-hos 8000.

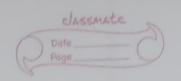
· allow - none = True: - allows · None' valley

the XML- free protocol does not allow None now values (null) to pe be passed between dient & sorver This makes server more no bust under conditions like:-· If ever occurs and function de ides to return none. · If you excland this application Register the factorial functions

so dients can prossy call prosed. Pactorial () - rpc dient py semotely using prosey. Factorial () when the server starts it should know which Hundions it should allow dients to use. which it server would nt know what to do when the dient · Server forever of & keeps the sorver · alive and listening for dient requests.

4) Run the server in a tex background thread

sorver - thread Et - ...



Greates a new deamon thread to sun the server function. · A deamon thread stops when the main program ends. . This allows the main program A A thread is a independent sequence Of instruction that can be threads within the same process. rpc - dient .py:i) imports the xmlxpc dient module

roug = xmlrpc. client. Server prusey (...)

- creats a prosey object.

- This objects lets you call semote

functions (on the server) as if they were local.

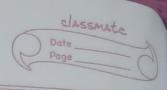
3) Main function: i) fasks the user to injust a number. iii) Converts in put to int.

iii) Calls the server function: proc. factorial

- Bends the request to

Server.

- Server computes and returns the result. iv) Displays the result.



(vi) Asks the user if they want to continue.

Real life applications:

Banking Systems

Banking Systems

Banking Systems

Banking Systems

Calculate use centralized servers to calculate

counts, (redit scores de

Clients (ATIMS, apps, websity) send

data to get results instally

2) Multiplayer games.
3) Any apps.
4) Online tax portals.

constitution and the second